Amendments to the Specification:

Please amend the paragraph beginning on page 2 of the application, at line 19, as follows:

BPbots can be used to describe the business process behavior at different levels. The key to ~~undeerstanding~~ understanding how BPbot enable autonomic ~~autonomoic~~ behavior is the notion of BPbot composition. BPbot communities can be composed in a hierarchical fashion or with independent BPbots, which work collectively towards achieving the targeted business goal. BPbots being the fundamental autonomic solution components to be wired together representing an autonomic business process will necessitate the overall construct to be a BPbot, as well. This implies that the autonomic business process solution is a BPbot, which consists of a managerial unit and execution modules. From this perspective, the execution modules are BPbots. If the BPbot community is nested, the managerial module (which can be a BPbot, as well) represents the major point of control in the BPbot community. In a non-hierarchical community, the BPbots are going to be interacting collaboratively to achieve the targeted business goal.

Please amend the paragraph on page 4 of the application, at lines 1-9, as follows:

Adaptive Business objects (ABO) model the executable image of business artifacts. ABOs can be viewed as a model for business records, containing data pertinent to the business process and handed from task to task along the business process. An ABO serves as a dynamic aggregator of distributed business data corresponding to a business artifact and models the life cycle and associated behavior of the business artifact. Augmented finite state machines define ABO behavior. Business events are processed by an ABO based on its state and may trigger state transitions. As part of a state transition, an ABO may execute commands to effect the business environment ~~enfironment~~.

Please amend the paragraph on page 4 of the application, at lines 10-16, as follows:

Workflows model a sequence of activities. An activity is a unit of collaboration. Typically, a set of activities performed by organizational role players ~~constitutesd~~ constituted a business task (or respectively, a particular state of the ABO). Flow models are ~~use~~ used to specify what the activities are, who are performing them, and the control flow between the activities. The activities themselves are defined using augmented finite state machines much like the ABO. But unlike ABOs, activities have no data content.

Please amend the paragraph beginning on page 4, at line 22, as follows:

Wiring these three components in the right way will enable an IT system, which will exhibit the behavior of the business process. At design time, the components will contain customized models such that the overall behavior of the solution represents the business process behavior. The created IT system, which will execute the models and thereby host the business process, should be designed to allow for changes. This does not mean, however, that changes due to changing business conditions or changing business goals can be managed by the system itself. It will always require human intervention to reconfigure the system manually ~~manuallyl~~. To create a truly adaptive system, which is self-managing (and thereby adaptive to changing business conditions) to limit the manual changes to a minimum, the system needs to consist of autonomic solution components. The entire solution for a given business process consists of autonomic components (or BPbots) and an autonomic component or BPbot itself. By the same token, the set of enterprise wide business process should consists of autonomic business processes and be an autonomic business process system itself. BPbots reflect this hierarchical ~~hierarichal~~ structure in its ~~it~~ architecture.

Please amend the paragraph on page 7, lines 3-18, as follows:

Referring now to the drawings, and more particularly to Figure 1, there is shown the architecture of a BPbot at deploy time. As mentioned, the BPbot consists of a managerial module 10 and an execution module 12. The managerial module 10 includes at least one manager service definition 101, which describe the basic capabilities of the managerial module. The manager service is capable of creating execution plans for the execution module. The managerial module 10 also includes a plurality of conversation policies (CPs) 102, which are descriptions of communication policies between BPbots, and a plurality of knowledge services 103, which are descriptions of services needed to support the creation of execution plans. The execution module 12 includes a plurality of execution service definitions 121, which describe the basic capabilities of the execution module. The manager service definition 101 and the execution service definitions 121 are linked, as indicated ~~indicted~~ by the double headed arrow. The execution module 12 also includes a plurality of behavior modification service definitions 122, which can be modified by the manager service definition 101 as may be required by changing business conditions.

Please amend the paragraph beginning on page 7, at line 28, as follows:

Figure 2 depicts an example scenario of the internal BPbot execution at run-time. The BPbot manager service run-time 14 exposes a sensor channel 21 through which it receives a message from the outside. The manager service 101 accesses knowledge resources 16. In this example, the knowledge resources are public resources (as opposed to the local conversation policy resources, which are private to the BPbot). The manager service 101 uses the decisions provided by the knowledge service 16 to create an execution plan for the execution module. This execution plan is conveyed to the modification service run-time 18, which stores the execution plan. The manager service 101 sends a message to the execution service to 121 execute the next step. The execution run-time services 17 expose a sensor 22 through which it receives the message from the manager service 101. The execution services 121 checks with the modification service 18 to see whether the manager service 101 has requested a change of execution. The modification service 18 responds by deploying the appropriate execution script based on the execution plan designed by the manager ~~manger~~ service 101. The execution is thereby changed. The execution service 121 exposes an effector 23, though which the execution result is sent to the manager service 101. The manager service 101 consults the conversation policy 102 to determine the next state in the conversation with another BPbot and determines the required communication protocol. The manager service 101 exposes [[a]] an effector 24 and sends the message out.

Please amend the paragraph which begins on page 8 at line 21 as follows:

Figure 3 illustrates one example of a concrete Bpbot implementation ~~implementatin~~ allowing for parametric changes to a process flow. In this BPbot implementation, the manager ~~manger~~ service 101 of the managerial module 10 is realized as a simple message driven JavaBean, which enables a publish-subscribe mechanism to send and receive messages. Furthermore, the manager service 101 has the capability to create rule sets and deploy these rule sets to the ABLE Rule Engine 122′. ABLE is a Java™ framework, component library and productivity tool kit for building intelligent agents using machine learning and reasoning. The ABLE framework provides a set of Java™ interfaces and base classes used to build a library of JavaBeans called AbleBeans. The library includes AbleBeans for reading and writing text and database data, for data transformation and scaling,[[,]] for rule-based inferencing using Boolean and fuzzy logic, and for machine learning techniques such as neural networks, Bayesian classifiers, and decision trees.

Please amend the paragraph beginning on page 11, at line 13, as follows:

Figure 5 illustrates the application of the invention to the retail industry for a simple replenishment process. A retail store receives point ~~pont~~ of sales events 51 every time a product is sold at the counter. Radio frequency identification (RFID) events 52 notify the arrival of goods at the backroom. The retail store also receives advanced shipment notices 53 about goods shipped out to be received soon. The retail store BPbot 54 evaluates the incoming events and determines if there is a potential out of stock situation. The retail store BPbot 54 consists of two BPbots 541 and 542(similar to Figure 4), one determining potential out of stock situations, and the other handling the out-going shipment orders. Assume that the two internal BPbots have determined an out-of-stock threat due to unexpectedly strong sales for a product. This event is sent to the BPbot manager 543, which sends the messages to the Retail Headquarters (HQ) BPbot 55. The HQ BPbot 55 has visibility over the entire supply chain and more knowledge to ascertain the urgency of the situation. The managerial module 553 of the BPbot 55 receives the message and initiates a process in one of the internal BPbots 551 or 552 to elucidate the situation. Finally, the HQ BPbot 55 creates an execution plan, which is sent to the distribution center (DC) BPbot 56. The DC BPbot 56 receives the execution plan from the HQ BPbot 55 and initiates the execution by applying the procedure as described above. Due to the urgency of the out of stock situation, the execution plan requires a one time expedited delivery to the retail store. The rules for execution are deployed to the modification service of the DC BPbot 56, and the process is changed accordingly. The expedited shipment order is created and sent to the managerial module 563 of the DC BPbot 56. The managerial module 563 sends a replenishment order for the product to the supplier 57. The managerial module 563 unit also notifies the Retail Store BPbot 54 of the expedited shipment and its expected arrival date (Advanced Shipment Notice 58).